Tier2 Pilot on the Illinois Campus Cluster

The Tier2 Pilot project was established to prove that it is possible to remotely run jobs from an existing Tier2 on the Illinois Campus Cluster (ICC) called "Taub". The goal was the show a potential method using existing infrastructure, networking, nodes and disk space. No funding was initially available for this project

The plan was to use the Condor Flocking feature to submit the jobs from an existing Tier2 cluster to the ICC. When two or more Condor masters flock to/from each other, jobs submitted by a host on the "Flock From" Condor instance will first attempt to match a node on the local instance. If no match is found, then Condor will attempt to match a node on the remote instance. If a match is found, the job will then "Flock To" the remote instance and run on a remote node.

By replicating various components of a local Tier2 onto the ICC, and by creating appropriate entries in schedconfig on the Panda queue used by the submission of jobs, it is possible to flock pilots to the ICC, and subsequently run production and analysis jobs.

In Phase 1 of the pilot program, the IllinoisHEP Tier3gs was used as the Tier2. Though small compared to true Tier2 sites, such as MWT2, functionally it is equivalent. In Phase 2 of the pilot, the plans are to use MWT2.

Existing Infrastructure Used in the Pilot

Pilot nodes

To aid in this pilot, the College of Engineering provided four nodes which are currently part of the "Taub" campus cluster along with some disk space on the existing GPFS DDN. These nodes are HP machines (model unknown) with dual core Intel Westmere processors (HT disabled) providing 12 jobs slots in total. Each node has 24GB and a 160GB local hard drive, two 1Gb NICS and a 40Gb InfiniBand interface.

The nodes are named taub205, taub206, taub207 and taub208.

These nodes PXE boot their system image into a RAM disk set at 1GB. An additional 1GB of memory is also used by the GPFS file system for local caching. As such, the pilot nodes are a bit weak in the total memory.

The local disk was to be used for many functions. Swap, CVMFS cache, local condor configuration files and local as well as the WN scratch space (with pCache). Since 160GB would be too tight, HEP loaned three 1TB SATA-300 drives to the pilot project. Unfortunately the HP nodes can only accommodate one local hard drive.

Taub Networking

All the worker nodes on the Taub cluster are isolated on a private non-routing network which uses the InfiniBand network. This is also true for the nodes provided for the pilot. Normally this would not be a problem. However a NAT router was not available to allow access by the worker nodes to the public network. It was necessary to first connect and assign the pilot nodes to the public network via one of the 1GB NICs. This was an easy task; however the campus firewall became an issue.

The public IPs for the pilot nodes are (they are not registered in DNS)

taub205	192.17.34.58
taub206	192.17.34.59
taub207	192.17.34.61
taub208	192.17.34.62

The Taub cluster public IP space is located on the campus network which sits behind a campus firewall. The policy for this subnet on the campus firewall was set to “mostly closed”. As a result, nearly all ports are blocked at the campus entry. The IllinoisHEP Tier3gs resides outside the campus on the Campus Research Network (CRN). As such, all traffic between the ICC and IllinoisHEP would traverse the campus firewall. The same will be true when MWT2 is ultimately used in the pilot. By default Condor needs to have ports 9418 (Collector), and those specified between LOWPORT and HIGHPORT in the shared condor_config open between the submit host and remote Condor instance. To get around the campus firewall, it was necessary to use Condors SHARED_PORTS and Condor Connection Block (CCB) features.

IllinoisHEP Tier3gs

The existing IllinoisHEP Tier3gs was used in the pilot as a stand-in for the MWT2.

The IllinoisHEP system consists of a CE/gatekeeper (osgx0), a dCache based SE with 125TB of storage (SRM osgx1, doors, osgx1/2/3), Gums (osggums), Squid (osgsquid) and 19 worker nodes (wn00 through wn18).

The local scheduling system is Condor 7.6.1 with a condor master (osgcondor)

All endpoints are registered in the Tiers of Atlas (ILLINOISHEP_USERDISK, etc).

Panda is used as the grid queuing system with queues IllinoisHEP, ANALY_IllinoisHEP and MP_IllinoisHEP. Two new queues, IllinoisCC and ANALY_IllinoisCC were created within the IllinoisHEP site for use in this pilot.

ICC Job Environment

It is mandatory that the job environment on the IllinoisHEP Tier3gs be replicated on the ICC and available via the same paths. This is due to the fact that the CE specifies the path to the $OSGAPP directory and various schedconfig variables in the Panda queue define the paths to WN Client and WN Scratch.

To accomplish this, three areas are replicated via rsync from IllinoisHEP. They are $OSGAPP, WN Client and the Certificate Authority (CA). Also CVMFS, required RPMS and some softlinks were installed on every pilot worker node.

These three areas are replicated via rsync by setting up an rsync server on osgx0.

```
[root@osgx0 ~] # more /etc/rsyncd.conf
log file = /var/log/rsyncd.log


[WNClient]
   path    = /home/osg/WN
   comment = OSG WNClient
   read only = yes
   uid = root
   gid = root
   hosts allow = 192.17.34.40 128.174.118.140

[OSGAPP]
   path    = /home/osgstore/app
   comment = OSG Application tree
   read only = yes
   uid = root
   gid = root
   hosts allow = 192.17.34.40 128.174.118.140

[CA]
   path    = /home/osg/CA
   comment = OSG Certificate Authority
   read only = yes
   uid = root
   gid = root
   hosts allow = 192.17.34.40 128.174.118.140
```

The Taub management node (192.17.34.40) replicates the above to a GPFS file system with the following rsync commands executed as a cron job.

```
rsync --delete -aq osgx0.hep.uiuc.edu::WNClient/ /home/osg/WN
rsync --delete -aq osgx0.hep.uiuc.edu::OSGAPP/ /home/osgstore/app
rsync --delete -aq osgx0.hep.uiuc.edu::CA/ /home/osg/CA
```

CVMFS was installed on every worker node. Since a local squid was not available on the ICC network, the IllinoisHEP squid (osgsquid) was used.

The standard set of RPMS needed by the Atlas releases, pilot, etc must be installed on every pilot worker node.

The path /etc/grid-security/certificates is soft linked to the replicated CA area.

```
# mkdir /etc/grid-security
# ln –s /home/osg/CA /etc/grid-security/certificates
```

The WN Scratch space is setup on the local scratch disk and soft linked from the same path as used on IllinoisHEP.The local scratch disk on the pilot nodes is mounted at /scratch.local. To tie into the same logical path to the WN Scratch space as is used on IllinoisHEP, the following was executed on each worker node

```
# mkdir /scratch.local/osgtmp
# chown daemon:usatlas /scratch.local/osgtmp
# chmod 775 /scratch.local/osgtmp
# chmod +t /scratch.local/osgtmp
# ln –s /scratch.local/osgtmp /home/osgstore/tmp
```

It was also necessary to create the usatlas accounts and groups on the ICC. The same UID/GID pairs from IllinoisHEP were used on the ICC to ease in the testing It is also necessary that the HOME area for each account on the ICC be identical as those used on IllinoisHEP. The following were created

Users:

UID	Name	Home
2001	usatlas1	/home/osguser/usatlas1
2002	usatlas2	/home/osguser/usatlas2
2003	usatlas3	/home/osguser/usatlas3
2004	usatlas4	/home/osguser/usatlas4

Groups:

GID	Name
2000	usatlas
2001	usatlas1

2002	usatlas2
2003	usatlas3
2004	usatlas4

By using the same UID/GID for each user and group, the replicated files will have the same ownership and permissions on the ICC as given on IllinoisHEP.

Condor Flocking Configuration

Since IllinoisHEP uses Condor as its local scheduler, and MWT2 is switching to Condor, it made sense to try and use the flocking feature built into Condor. A few issues came to light when trying to setup flocking to the ICC

The first issue was to create an ICC condor master to use in the pilot. This was done using one of the pilot nodes (taub208). Taub208 was given a public IP (192.17.34.62), but not registered in DNS. The shared condor_config was taken from the IllinoisHEP condor master, with appropriate changes made for use on the ICC networks, etc.

Normally, flocking is established between two Condor instances by setting the FLOCK_TO and FLOCK_FROM values appropriately. However, due to the restrictions enforced by the firewall on the ICC, additional steps were needed.

On the taub208 it was necessary to enable SHARED_PORTS and the CCB.
This was accomplished by adding the following to the shared condor_config.
Port 5988 is open in the campus firewall for the ICC.

```
# Enable shared ports via port 5988
SHARED_PORT_ARGS = -p 5988
COLLECTOR_HOST = $(CONDOR_HOST):5988?sock=collector
USE_SHARED_PORT = TRUE

# CCB settings
CCB_ADDRESS = $(COLLECTOR_HOST)
PRIVATE_NETWORK_NAME = campuscluster.illinois.edu
```

Every pilot node on the ICC must run the SHARED_PORT daemon.

```
DAEMON_LIST =  SHARED_PORT, <other daemons>
```

To enable flocking, the following entries are made to each condor master.

On ICC

```
FLOCK_FROM = osgcondor.hep.uiuc.edu
ALLOW_READ = *.hep.uiuc.edu, *.campuscluster.illinois.edu
ALLOW_WRITE = *.hep.uiuc.edu, *.campuscluster.illinois.edu
```

On IllinoisHEP

```
FLOCK_TO = 192.17.34.62:5988?sock=collector
ALLOW_READ = *.osg.hep.uiuc.edu, *.hep.uiuc.edu, 192.17.34.*
```

The following should be set on both Condor masters so that jobs will run as the same user on the remote instance (ie usatlas1 on IllinoisHEP maps to usatlas1 on ICC)

```
UID_DOMAIN           = USAtlas
TRUST_UID_DOMAIN = True
```

Panda Queues and schedconfig

It was necessary to create two new Panda queues to be used in the pilot. This allowed changes to the schedconfig parameters that will be unique to the ICC. For example, since the ICC does not have direct access to the IllinoisHEP SE, dcap cannot be used and the “copytoolin” must be set to “None” so that stageins will use “copytool” which is set to “lcg-cp2” for the existing IllinoisHEP queues.

Two new queues were created, IllinoisCC-condor and ANALY_IllinosCC-condor.

The queue configuration files were copies of the existing queues on IllinoisHEP, with the necessary changes to define a queue named IllinoisCC or ANALY_IllinoisCC.

The one major difference between the HEP and CC queues are in the JDLConfig. The CC queues need to specify various “Requirements” in the condor submit to allow the jobs to flock properly to the ICC. This is done by modifying the schedconfig parameter “globusrsl”

```
globusrsl      = (condorsubmit=('+IsCCJob' TRUE)('+IsSPJob' TRUE)('+IsProdJob' TRUE)('priority' 0)('Transfer_Executable'
TRUE)('When_To_Transfer_Output' ON_EXIT)('Should_Transfer_Files' IF_NEEDED)('Requirements' 'FileSystemDomain !=
\\\"\\\"'))
```

When a job is submitted via one of the CC queues, condor must transfer the submit control and executable file from IllinoisHEP to the ICC and any condor output files (stdout, stderr) must be transferred back from the ICC to IllinoisHEP. The FileSystemDomain requirement is needed to override the default behavior to only run the jobs on the file system from the “submit” node.

The classads, IsCCJob, IsSPJob and IsProdJob (or IsAnalyJob) are used within the IllinoisHEP Condor shared condor_config to restrict which job slots a job can “match”.

Job Flow With Flocking

The flow of a job submitted by Panda to a remote instance via flocking varies only slightly from the flow to a local instance. Here is a brief description on the job flow from Panda submission to completion.

When a CC queue is set to “brokeroff” or “online”, Panda will submit a proto pilot job via Glidein to the IllinoisHEP gatekeeper which is “osgx0.hep.uiuc.edu”.

Since the proto pilot is submitted via the CC queue, the Condor requirements from the “globusrsl” are appended into the control file. The Condor master on IllinoisHEP will first try to “match” the job to a slot on the IllinoisHEP cluster. If a match does not occur or no job slots are available, Condor will then attempt to “match” the job to a job slot on a flocked Condor. The “IsCCJob” value is used within IllinoisHEP to prevent a match to a IllinoisHEP job slot, and force a match to an ICC job slot.

With a match found on an ICC job slot, the submit control file and “Executable” (the proto pilot python script, is transferred to the ICC Condor master. The proto pilot then begins to run on a job slot assigned to one of the pilot worker nodes. Since the UID_DOMAIN is set to the same value on IllinoisHEP and the ICC (USAtlas), the pilot will run as the same user on the ICC as that on IllinoisHEP (usatlas1).

Since the job environment on the ICC is identical to that of IllinoisHEP, the proto pilot is able to find all of the components it needs to execute such as the $OSGAPP tree, Atlas releases (via CVMFS), the WN Client software stack and the WN Scratch working area.

Everything which follows is identical to a job which is run on the local instance.

Once the job completes, the condor logs are transferred back to IllinoisHEP. Since all datasets, logs and other Atlas output are transferred to/from the SE via the “copytool”, the amount of data moved between the two Condor instances is minimal.

Phase 1 Complete

The above work was first worked out in a Phase 0.5 between the IllinoisHEP Tier3 and the IllinoisHEP LX cluster. The transition to Phase 1 using the ICC as the target for flocking was relatively easy. Only the firewall issues needed to be resolved.

The Panda queues, IllinoisCC and ANALY_IllinoisCC were put into a “brokeroff” state. Test production and analysis jobs were run from October 5, 2011.through October 12 to make certain there were no “hidden” issues that might arise.

On October 12, 2011 at 4PM, the IllinoisCC queue was put into an “online” state. Production jobs began to run immediately. Two days later, ANALY_IllinoisCC was also put into an “online” state and user analysis jobs began running for the first time.

Since then, several thousand production and analysis jobs have run on the ICC via the gatekeeper at IllinoisHEP and using the IllinoisHEP SE.

http://panda.cern.ch/server/pandamon/query?joCb=*&type=&days=1&jobsetID=any&jobStatus=&site=IllinoisCC&plot=no

http://panda.cern.ch/server/pandamon/query?job=*&type=&days=1&jobsetID=any&jobStatus=&site=ANALY_IllinoisCC&plot=no

Phase 2 Next….

The next step will be to repeat this procedure using MWT2 as source Tier2.

The best approach will be to rsync $OSGAPP, WN Client and WN Scratch from MWT2 to the locations where MWT2 has them located. Also, the UID/GID and HOME for the usatlas users and groups need to be changed to match those in use at MWT2. It would also be best to create to new Panda queues, something like MWT2_ICC and ANALY_MWT2_ICC. The schedconfig parameters on these queues can then be set with the appropriate values for “globusrsl”, etc without interfering with those queues in production. Also, the necessary changes to the MWT2 Condor master configuration to enable FLOCK_TO the ICC need to be implemented.

The firewall at MWT2 and local site mover will most likely present some new challenges to be worked out in this transition. Also, the networking connection between Illinois, UChicago and Indiana University will be somewhat tested as a result of this pilot.